

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería Informática**

# **TRABAJO FIN DE GRADO**

**Optimización del itinerario de servicios para el  
desempleado**

**Sergio Bravo Santos**

**Tutor: Daniel Hidalgo Martínez**

**Ponente: Gonzalo Martínez Muñoz**

**Septiembre 2019**



# **Optimización del itinerario de servicios para el desempleado**

**AUTOR: Sergio Bravo Santos**

**TUTOR: Daniel Hidalgo Martínez**

**Dpto. Ingeniería Informática  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Septiembre de 2019**





# Resumen (castellano)

Accenture, junto con la Comunidad de Madrid y otras empresas están desarrollando diversos proyectos de valor para mejorar la inserción de los demandantes de empleo.

En concreto, uno de los proyectos trata sobre la segmentación de los demandantes en distintos grupos de empleabilidad y la simulación de la mejor acción posible a tomar (NBA) para que el desempleado encuentre trabajo con mayor probabilidad lo antes posible.

Uno de los principales problemas que tiene el sistema NBA mencionado anteriormente es que solo busca elaborar una acción, la mejor posible, no un itinerario completo. Además, el método de NBA no tiene en cuenta los horizontes temporales, es decir, no discierne entre la duración de cada una de las acciones, dando lugar a situaciones en las que se puede elegir una acción que potencialmente durará 12 meses, en vez de elegir otra acción que tenga una mejor relación en cuanto a calidad/duración se refiere.

En este trabajo de Fin de Grado se pretende buscar una alternativa a dichos problemas y se aumentará la calidad del servicio ofertado mediante un sistema de simulaciones más adaptado a la necesidad de los demandantes.

El sistema de simulaciones tendrá como núcleo principal el método búsqueda en árbol de Montecarlo. El método de Montecarlo ha sido enteramente diseñado desde cero, aunque respetando la metodología del método original, pero realizando ciertas modificaciones que inducirán a un mejor funcionamiento y una mejor adaptabilidad al problema surgido. Durante la lectura de este documento se podrá ver en detalle dichos cambios realizados junto con la explicación de estos.

A lo largo del documento se verá reflejado en detalle el motivo del problema, el trabajo de investigación, el estado del arte, el diseño de la solución, el desarrollo del proyecto, los resultados obtenidos y por último las conclusiones obtenidas a partir de dichos resultados.

# Abstract (English)

Accenture, in collaboration with the Community of Madrid and other companies are developing together important projects to improve employment insertion for the unemployed.

Specifically, one of the projects is about the segmentation of the demandants into different employability clusters and the simulation of the next best possible action (NBA) to be taken for the unemployed in order to find work as soon as possible.

One of the main problems that the NBA system has, is that it only search on possible action, the best one, not a complete itinerate of actions. Moreover, the NBA method do not take into account the time horizons, is it to say, it does not discerns into the duration of the actions, giving place of situations where a really long time-consuming action could be chosen over one action that may seem a little bit less-effective, but is way much shorter, so it is worth to choose that one in terms of the effectiveness/duration ratio.

In this final Project, it is pretended to find an alternative solution for those problems. It is also our task, to improve the quality of the given service using a simulation system better adapted to the necessity of the demandants.

The simulation system will have as its main core, the Monte Carlo's Search Tree method. The Monte Carlo's method has been entirely implemented from scratch, although it respects the methodology of the original one, but with some modifications which will induce into a better performance and a better adaptability to the problem that arose. During the reading of this document it can be seen in more detail those changes, with an explanation of what they are used for attached.

All along this document it will be seen in detail the reasons to make the project, the investigation work to find a solution, the art state, the solution design, the development steps taken, the results obtained and finally the conclusions induced from those results.

## **Palabras clave (castellano)**

Accenture, Comunidad de Madrid, Desempleados, Perfilado, Simulación, NBA, horizonte temporal, acciones, relación calidad/duración, método de búsqueda en árbol de Montecarlo.

## **Keywords (inglés)**

Accenture, Community of Madrid, Unemployed, Segmentation, Simulation, NBA, Time horizon, Actions, effectiveness/duration ratio, Monte Carlo search tree method





## ***Agradecimientos***

*Quiero agradecerle este trabajo a mi familia que siempre me ha motivado a seguir estudiando y a formarme en el campo de la informática, sin ellos no estaría donde estoy.*

*A mis amigos por apoyarme, aportar ideas y ayudarme a aguantar la presión en momentos imposibles.*

*Por último mencionar a todo el equipo de Applied Intelligence de Accenture que me ha echado una mano siempre que ha sido posible y me ha guiado durante el periodo en el que he estado con ellos.*



# INDICE DE CONTENIDOS

<b>1. Introducción</b>	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Organización de la memoria	2
<b>2. Estado del Arte</b>	5
2.1. Desarrollos previos del sistema integral de empleo	5
2.1.1. Registro	5
2.1.2. Perfilado	6
2.1.3. Simulaciones	7
2.1.4. NBA	8
2.1.5. Orientación	9
2.2. Arquitectura	9
2.2.1. Lenguaje utilizado	9
2.2.2. Librerías utilizadas	9
2.2.3. Google Cloud Platform	10
<b>3. Diseño</b>	11
3.1. Datos disponibles	11
3.1.1. Demandantes	11
3.1.2. Variables de evaluación de mercado	12
3.1.3. Coeficientes	12
3.2. Entrenamiento de perfilado	13
3.3. Solución analítica: Método de Montecarlo	14
<b>4. Desarrollo</b>	15
4.1. Extracción y manejo de datos y función de perfilado	15
4.2. Google Cloud Platform	17
4.3. Método de Montecarlo	17
4.3.1. Árbol	18
4.3.2. Selección	20
4.3.3. Expansión	21
4.3.4. Puntuación	21
4.3.5. Simulación	23
4.3.6. Back-propagation	24
4.3.7. Obtener el mejor hijo	24

4.3.8. Umbral temporal .....	25
4.3.9. Últimos pasos .....	26
<b>5. Integración, pruebas y resultados .....</b>	<b>27</b>
5.1. Elección de demandantes para la validación .....	27
5.2. Elección de hiper parámetros .....	27
5.3. Resultados .....	31
5.3.1. Flujo de demandantes .....	32
5.3.2. Varianza de probabilidades .....	34
5.3.3. Número de acciones por itinerario .....	35
5.3.4. Duración de itinerarios .....	36
5.3.5. Frecuencia de itinearios .....	37
<b>6. Conclusiones y trabajo futuro .....</b>	<b>39</b>
6.1. Conclusiones .....	39
6.2. Trabajo futuro .....	40
<b>Referencias .....</b>	<b>41</b>
<b>Glosario .....</b>	<b>45</b>
<b>Anexos .....</b>	<b>I</b>
A. Método de Montecarlo .....	I

## INDICE DE FIGURAS

<b>Figura 4-2-1:</b> Esquema conceptual de la puntuación .....	22
<b>Figura 5-2-1:</b> Comparación de hiper parámetros para los demandantes que cambian del grupo B a un grupo superior .....	29
<b>Figura 5-2-2:</b> Comparación de hiper parámetros para los demandantes que cambian del grupo B a un grupo superior .....	29
<b>Figura 5-2-3:</b> Comparación de hiper parámetros para los demandantes que cambian del grupo B a un grupo superior .....	30
<b>Figura 5-2-4:</b> Comparación de hiper parámetros para la variabilidad en la probabilidad de salir del desempleo .....	30
<b>Figura 5-3-1:</b> Flujo de demandantes del grupo B a grupos superiores para NBA y Montecarlo ...	32
<b>Figura 5-3-2:</b> Flujo de demandantes del grupo C a grupos superiores para NBA y Montecarlo ...	32
<b>Figura 5-3-3:</b> Flujo de demandantes del grupo D a grupos superiores para NBA y Montecarlo ...	33
<b>Figura 5-3-4:</b> Diagrama de cajas y bigotes de varianza de la probabilidad por grupo para NBA y Montecarlo .....	34
<b>Figura 5-3-5:</b> Diagrama circular de número de acciones por cada itinerario para NBA y Montecarlo .....	35
<b>Figura 5-3-6:</b> Diagrama circular de la duración de los itinerarios para NBA y Montecarlo .....	36
<b>Figura 5-3-7:</b> Itinerarios más frecuentes para NBA y Montecarlo .....	38

## INDICE DE TABLAS

<b>Tabla 2-2-1:</b> Librerías utilizadas .....	9
<b>Tabla 3-1-1:</b> Variables de los demandantes .....	12
<b>Tabla 3-1-2:</b> Variables del fichero de coeficientes .....	13
<b>Tabla 4-3-1:</b> Campos del árbol .....	19
<b>Tabla 4-3-2:</b> Puntuaciones iniciales .....	21



# 1 Introducción

---

## 1.1 Motivación

En segundo de carrera asistí a una charla sobre redes neuronales y ciertos proyectos que trataban de imitar el comportamiento del cerebro humano, desde ese momento cogí un gran interés acerca de todas las posibilidades que nos ofrece la inteligencia artificial y supe que ese era el camino que quería seguir.

La idea de hacer un TFG que tenga que ver con el mundo del machine learning y el análisis de datos era clara, sin embargo, faltaba encontrar un proyecto que me permita no solo aprender sobre el campo, sino hacer algo útil que sirva en el mundo real.

Por esta razón, contacté con Dani Hidalgo (responsable del sector público para Accenture Applied Intelligence) para ver qué posibilidades tenía. Dani, aprovechando la colaboración de Accenture con la Comunidad de Madrid, me ofreció la posibilidad de trabajar en una solución que podría ser relevante a futuro para el cliente, la cual trataba de perfilar y orientar profesionalmente a los desempleados mediante el uso de tecnologías diversas.

El objetivo de la solución mencionada era tratar de mejorar el sistema de predicción de acciones futuras para orientar al demandante. Para ello, se pensó en incorporar un sistema que prediga que itinerario podría ser el más indicado a realizar, de manera que el demandante consiga mejorar su situación de desempleado y se incorpore lo antes posible en el mundo laboral.

Una vez sentadas las bases y sabiendo el principal problema, solo queda pensar en una solución que consiga mejorar el modelo actual y que se ajuste a nuestras necesidades.

## 1.2 Objetivos

La ambición que se tiene con este proyecto es mejorar el actual sistema de recomendación de acciones.

Actualmente se cuenta con un sistema que predice una acción. La idea es hallar la forma, no solo recomendar una acción posible, sino implementar un itinerario entero de



acciones a tomar que garantice una mejor probabilidad de que el desempleado encuentre trabajo lo antes posible.

La solución ofertada permitía además seguir la temporalidad de las acciones, no como en el sistema actual, que ofrece una sola acción a tomar sin importar los meses que tarde el demandante en adquirir dicha habilidad.

Por último, el objetivo principal del proyecto es proponer un sistema que mejore los resultados del anterior NBA, y que sea capaz de analizar diferentes opciones que el algoritmo NBA no analizaría si dicho algoritmo fuera iterado.

En conclusión, se quiere un sistema más robusto y fiable que asegure que el desempleado encuentra un camino a seguir que mejore su situación en la mayor medida de lo posible.

### ***1.3 Organización de la memoria***

Este documento se divide en 6 capítulos diferentes:

En la introducción se habla sobre las causas que motivaron la realización del proyecto del que vamos a hablar y de los objetivos que se pretenden alcanzar.

En el estado del arte se dará una visión sobre el estado del desarrollo, es decir, hablaremos de los proyectos de los que partimos, a parte de tratar las tecnologías con las que desarrollaremos nuestro proyecto junto con la explicación de estas.

En el capítulo de diseño, se hablará de los datos disponibles y, contando con la información que se sabe, y los datos mencionados, como se llegó a la solución que tratará de solventar nuestro problema.

En el capítulo de desarrollo se podrá ver el flujo de trabajo que se ha seguido y como se ha implementado la solución del problema.

En Integración, pruebas y resultados, se encuentran las gráficas que muestran los resultados del sistema implantado y se compararán los resultados obtenidos con el sistema antiguo para validar que el modelo funciona correctamente.

En el último capítulo se obtendrán unas conclusiones a partir de la información tratada en el resto del documento, y se finalizará el trabajo comentando futuros cambios que podrían implementarse para la mejora del sistema.



## 2 Estado del arte

---

Para una mejor comprensión de este TFG, es necesario proporcionar una visión de las tecnologías utilizadas para la realización del proyecto. De esta manera, en este capítulo se situará al lector en un contexto que ayude a comprender los aspectos del proyecto.

Este capítulo se divide en los siguientes apartados: En primer lugar, en la sección 2.1 se hablará del contexto en el que se sitúa el sistema integral de empleo, entrando en detalles de todos los desarrollos previos de este y la situación en la que se encontraba durante el desarrollo de este trabajo. En la sección 2.2, se mostrará la arquitectura del proyecto, tanto los lenguajes utilizados como las librerías y todas las tecnologías implicadas para la realización de este.

### ***2.1 Desarrollos previos del sistema integral de empleo***

En esta sección se explicarán todos los desarrollos previamente realizados por el equipo de Accenture encargado del sistema integral de empleo.

Esta sección está enfocada a dar un contexto previo del sistema de empleabilidad, de manera que, al hacer referencia a cualquiera de los puntos tratados, se tenga una idea clara de lo que estamos hablando.

Cabe también mencionar que el sistema integral de empleo cuenta con otros proyectos que no se tratan en este TFG debido a la poca relevancia que tienen en correlación al proyecto que se va a desarrollar.

Dicho esto, a continuación, se van a ver punto por punto cual sería el flujo de un demandante dentro del sistema.

#### **2.1.1 Registro**

El demandante se registra como desempleado en la Comunidad de Madrid, para ello tiene que rellenar un formulario con datos sobre su demanda de empleo que más tarde se añadirán a la base de datos del sistema.

Los datos pedidos serán desde datos sociodemográficos como fecha de nacimiento, sexo, lugar de nacimiento... hasta datos de demanda acerca de que ocupaciones desea

aplicar, experiencia en dichas ocupaciones... Se hablará más en detalle sobre los datos del demandante en la sección de datos 3.1.

Por último, el demandante contará con la ayuda de un técnico que le podrá orientar a la hora de encontrar una ocupación utilizando para ello datos de la oferta y la demanda de empleos.

### **2.1.2 Perfilado**

Después del registro en la Comunidad de Madrid, se realizará un perfilado contando con los datos del demandante obtenidos (el formulario) en la anterior etapa.

El perfilado es uno de los pilares principales que sostienen el nuevo sistema integral de empleo.

Éste trata de separar en segmentos distintos a los diferentes demandantes según sus probabilidades de encontrar trabajo en los diferentes horizontes temporales (3, 6 y 12 meses).

Dichas probabilidades son calculadas aplicando una regresión logística sobre los datos disponibles mencionados anteriormente.

Una vez se obtengan las probabilidades de salir del desempleo de los demandantes, se pueden formar grupos de demandantes dependiendo de la probabilidad que tengan de salir del desempleo y de una serie de umbrales previamente calculados en el modelo.

Hay cuatro grupos de demandantes:

- ❖ En el primer grupo se incluirán a los demandantes cuya probabilidad de salir del desempleo en 12 meses no supere el umbral de dicho horizonte temporal, por lo tanto, estos se registrarán por la probabilidad que tengan de salir en 12 meses.
- ❖ El segundo grupo incluirá aquellos demandantes cuya probabilidad de salir en 12 meses si supere el umbral, pero que la probabilidad de salir en 6 meses no supere el umbral correspondiente a dicho horizonte temporal. Estos se registrarán por la probabilidad de 6 meses.
- ❖ El tercer grupo incluirá aquellos demandantes cuya probabilidad de salir en 12 y 6 meses si supere respectivos umbrales, pero la probabilidad de salir en 3 meses

no superará el umbral de este horizonte temporal. Se registrarán por la probabilidad con horizonte temporal de 3 meses.

- ❖ El cuarto y último grupo incluye los demandantes cuyas probabilidades superen todos los umbrales. También se rigen por la probabilidad de salir en 3 meses.

### **2.1.3 Simulaciones**

Una vez se ha perfilado al demandante, se pasará a la etapa de simulación para identificar las diferentes acciones a ofrecer desde el sistema de empleo.

Para entender el documento es necesario comprender que es una acción.

Una acción es una variable del demandante que puede ser modificada con el fin de cambiar una o varias de las características del demandante. Las acciones realizadas sobre un demandante pueden afectar a otras variables del demandante que dependen de dicha acción.

Continuando con el motivo de la sección, una simulación es un proceso en el cual, se trata de recrear un escenario en el que el demandante ha adquirido o mejorado una acción, de esta forma podemos ver si realizando de nuevo el perfilado, este ha mejorado su situación actual de empleo.

Es importante decir que, cuando se cambia una acción del demandante, dicha acción puede afectar a otras variables del demandante, variables que podrían afectar en mayor o menor medida a que el demandante mejore o empeore su perfil (tanto sus probabilidades en los horizontes temporales como el segmento al que correspondería).

Hay diferentes acciones a tomar según la situación y datos del demandante. Se realizará una simulación para cada acción posible, a continuación, veremos una lista de las acciones posibles a tomar:

- ❖ Aumentar de nivel de inglés.
- ❖ Aumentar el nivel de informática.
- ❖ Aumentar el nivel de formación.
- ❖ Solicitar una ocupación que no se ha solicitado.
- ❖ Aumentar la experiencia en 12 meses en una ocupación que haya sido solicitada mediante un plan activo de empleo.

Una vez realizada la acción dentro de una simulación, perfilamos al demandante para obtener su probabilidad y segmento simulados.

El objetivo de dicha etapa es analizar la mejora del demandante en caso de que se lleguen a realizar estas acciones en el mundo real.

#### **2.1.4 NBA**

Una vez realizadas las simulaciones, el último paso es la NBA.

Para entenderlo mejor, el NBA es un término aplicado al marketing que considera distintas acciones a tomar entre todas las posibles para un usuario concreto y elige de entre ellas la mejor.

Tal y como se aplica el término asociado al marketing, lo aplicaremos en el sistema integral de empleo antiguo. En este caso, de entre todas las acciones simuladas, se recomendará aquella que haya conseguido unos resultados mejores en el perfilado. En caso de que ninguna acción consiga un mejor resultado en el perfilado que la que consigue el demandante sin haber sido simulado, no se recomendará ninguna acción a tomar.

#### **2.1.5 Orientación**

Una vez realizado el NBA, mediante una serie de condiciones se averigua si el demandante que se está tratando necesita un periodo orientativo o no, y en caso de necesitarlo, cada cuanto tiempo tiene que realizar una orientación.

En ella, el demandante será guiado por un orientador que le mostrará su situación y le recomendará diferentes acciones a tomar según la situación del mercado y los datos obtenidos de su formulario, perfilado y NBA.

## 2.2 Arquitectura

### 2.2.1 Lenguaje utilizado

Para el desarrollo del proyecto he decidido utilizar Python por diversos motivos.

En primer lugar, es un lenguaje que se trabaja en el periodo universitario, parecía interesante utilizarlo en un proyecto real y aplicar los conocimientos adquiridos para la realización de un proyecto, que, en un futuro, podría utilizarse para el mundo real.

En segundo lugar, he trabajado anteriormente con Python aplicado a proyectos de análisis de datos y se puede decir que Python es un lenguaje ideal, ya que es ampliamente apoyado por la comunidad y se han creado librerías que te aportan funciones necesarias para el desarrollo del trabajo.

Por otro lado, Python es un estándar open-source, por lo que, en el futuro, podría ser aprovechado para desarrollar una solución para el cliente.

### 2.2.2 Librerías utilizadas

Librería	Descripción
<b>CSV</b>	Implementa clases para leer y escribir información en formato csv. Es muy útil para leer la información de los ficheros con los que contamos, ya que la mayoría son ficheros csv. Ver referencia [9]
<b>Pandas</b>	Implementa funciones de lectura que devuelven estructuras de datos fáciles de usar. En el proyecto en concreto se utiliza para leer los ficheros.xlsx. Ver referencia [10]
<b>Scipy-Stats</b>	Implementa funciones estadísticas y distribuciones de probabilidad. Para el proyecto se utiliza la función norm, que devuelve un valor normalizado mediante la distribución normal. Ver referencia [15]
<b>Numpy</b>	Implementa funciones de algebra lineal, un objeto array n-dimensional y diversas funciones que te permiten trabajar con dicho objeto. Este objeto es muy útil para trabajar con matrices muy grandes de datos de manera eficiente. Ver referencia [20]
<b>Time</b>	Implementa funciones relacionadas al tiempo. Se utiliza la función time para averiguar el tiempo de ejecución de un proceso. Ver referencia [24]



<b>Random</b>	Implementa funciones que generan números aleatorios para diferentes distribuciones. Ver referencia [16]
<b>Matplotlib.pyplot</b>	Implementa funciones que generan gráficas. Se utiliza esta librería porque contiene todos los tipos de gráficas requeridos. Ver referencias [30] y [32]
<b>Operator</b>	Implementa un conjunto de funciones de operadores de Python más eficientes. En el código se utiliza itemgetter para obtener un ítem concreto de una lista, de esta forma poder ordenarlo en función de un valor de la lista. Ver referencia [25]
<b>Itertools</b>	Implementa funciones para iteradores (Un iterador es un objeto cuyos elementos pueden ser recorridos), en el trabajo se utiliza groupby para agrupar las listas por valor. Ver referencia [27]
<b>Statistics</b>	Implementa funciones para calcular estadísticas matemáticas. Se utiliza mean para calcular la media de los datos en el módulo de gráficas. Ver referencia [21]
<b>Copy</b>	Implementa funciones que copian estructuras de datos por completo. Esto significa que el objeto original no tiene peligro de ser cambiado cuando se modifica la copia. Ver referencia [17]
<b>Math</b>	Implementa las funciones matemáticas definidas en c. En el código utilizamos sqrt para obtener la raíz cuadrada de un valor. Ver referencia [35]
<b>Codecs</b>	Provee acceso al registro interno de Codec. En el proyecto se utiliza esta librería para poder leer ficheros csv en utf-16. Ver referencia [18]

**Tabla 2-2-1: Tabla con las librerías utilizadas y su descripción.**

### 2.2.3 Google Cloud Platform

Google Cloud Platform provee un servicio en la nube con el cual se puede hacer uso de sus máquinas virtuales personalizables.

En el nuevo sistema se utiliza GCP por que no se contaba con memoria suficiente para cargar toda la tabla de demandantes en una matriz numpy.

Una vez finalizada la etapa de codificación, se utilizó dicho servicio para realizar más simulaciones simultáneamente, ya que se podían aumentar las CPU's del sistema. Esto permitió aumentar el número de pasos del método y de simulaciones (hiper parámetros del método de Montecarlo) ya que se redujo mucho el tiempo de ejecución al realizar varias simulaciones en paralelo. Esto último implicó unos resultados más satisfactorios.

## 3 Diseño

---

Una vez introducido tanto los conceptos como el estado del proyecto, ya se puede enfocar la atención en el motivo del trabajo, es decir, se contará más específicamente los detalles de la solución para optimizar el itinerario de servicios para el desempleado.

En este capítulo se entrará en detalle con la etapa de diseño del algoritmo. Para ello, en la sección 3.1, se hablará de todos los datos con los que se disponen para el proyecto, en la sección 3.2 se muestra cómo se ha diseñado el perfilado para el nuevo algoritmo, y por último, en la sección 3.3, se hablará de los motivos por los cuales se ha elegido el método de Montecarlo como una buena solución al problema.

### ***3.1 Datos disponibles***

Para la realización del nuevo sistema se ha contado con una serie de datos útiles para el desarrollo de esta.

En esta sección se definirán los datos que se utilizarán en el proyecto e introduciremos una descripción sobre los mismos.

#### **3.1.1 Demandantes**

En primer lugar, se cuenta con datos de los demandantes.

Esta información ha sido obtenida a partir de los formularios realizados en la etapa de registro (Ver sección 2.2.1).

La tabla de demandantes tiene más de 300 variables para cada uno, dando una precisión mayor a la hora del perfilado. Cada variable tiene un peso asociado, lo que hará que ésta afecte en mayor o menor medida al perfilado de cada demandante.

A continuación, se van a mostrar algunas de las variables más relevantes de los demandantes:

Variable	Descripción
Número de IPF	Identificador del demandante
Edad	Edad del demandante.
Género	Género del demandante.
Código de discapacidad	El código que represente a la discapacidad del demandante en caso de tenerla.
Nivel de Formación	Nivel superior de formación que el demandante ha alcanzado.
Nivel de Inglés	Nivel de inglés hablado por el demandante.
Nivel de Informática	Nivel de informática del demandante
Experiencia en Ocupación	Experiencia en meses que tiene el demandante en una ocupación en concreto.
Solicitud de Ocupación	Nos informa si el demandante solicita o no una ocupación en concreto.
Nivel de Educación	Máximo nivel de educación del demandante.
...	...

**Tabla 3-1-1: Tabla simplificada con las variables del fichero de demandantes.**

### 3.1.2 Variables de evaluación de mercado

También se cuenta con una serie de variables que capturan la evolución del mercado.

Las variables de evaluación del mercado son aquellas que relacionan directamente los contratos con respecto a la demanda. En otras palabras, estas variables ayudarán a ver la cantidad de contratos por cada distinta ocupación.

Estas son calculadas por un modelo predictivo desarrollado en el antiguo sistema de empleabilidad, y se utilizarán como input para el perfilado.

### 3.1.3 Coeficientes

Por otro lado, también se cuenta con las variables de coeficientes.

Estas servirán para darle mayor o menor importancia a las variables de demandantes a la hora de perfilar a este.

Las variables de coeficientes, al igual que las variables de evaluación de mercado serán calculadas por el modelo predictivo mencionado en la sección anterior.

La siguiente tabla muestra un ejemplo

Variables del demandante	3m	6m	12m
Variable 1	1.423	1.32	1.89
Variable 2	0.45	0.67	-0.3
...	...	...	...
Variable X	0.12	0.37	1.2

**Tabla 3-1-2: Tabla explicativa de la tabla de coeficientes. La tabla tendrá diferentes valores según los horizontes temporales.**

### ***3.2 Entrenamiento de perfilado***

Es cierto que uno de los desarrollos previos en el sistema integral de empleo antiguo era el perfilado. Sin embargo, este perfilado no era del todo óptimo, pues no estaba preparado para perfilar todos los demandantes de forma vectorial, es decir, simultáneamente.

Es por esto por lo que se ha creado la misma función de perfilado en el lenguaje Python.

A su vez, el método elegido como solución del problema utiliza el perfilado para el entrenamiento de este, así que utilizar una función implementada en Python resulta en un algoritmo más cohesivo, y óptimo.

El nuevo perfilado no es más que una optimización del perfilado anterior, lo único que se ha cambiado es el poder obtener todos los perfiles de los demandantes a la vez, lo que lo hace más óptimo que el anterior sistema.

Para el nuevo perfilado también utilizaremos las variables de mercado y los coeficientes de la misma manera.

Cabe mencionar que, los umbrales necesarios para saber a que segmento pertenece cada demandante son obtenidos a partir de los coeficientes y de las variables de mercado.

### **3.3 Solución Analítica: Método de Montecarlo**

Una vez definido el contexto del problema y todos sus detalles, hay que centrarse en cómo se va a solucionar.

Los primeros días del proyecto se han dedicado a buscar un algoritmo capaz de asegurarnos una buena solución. En el transcurso de la búsqueda se contemplan varias posibilidades, entre ellas redes neuronales, el algoritmo SVM y por último el algoritmo de Montecarlo.

Finalmente se eligió el algoritmo de Montecarlo por las razones mencionadas a continuación: (Referencias [1], [2], [3] y [4])

- ❖ Montecarlo, asegura por probabilidad una solución buena, es decir, que mejora la situación del desempleado.
- ❖ La cantidad de itinerarios posibles es muy amplia, queriendo decir que recorrer todas las posibilidades del árbol es completamente inviable, es por ello por lo que usar un método estadístico es buena idea.
- ❖ No genera overfitting, cuantas más simulaciones se generen, mejor es el resultado. En este caso, pienso que esto es muy bueno, ya que para casos concretos en los que los resultados no fueran idóneos, se podrían realizar más simulaciones, asegurando un resultado mejor.
- ❖ El método de Montecarlo funciona bien en juegos como el tres en raya. De alguna manera, el problema al que nos enfrentamos se parece al problema del tres en raya, pues se trata de ver que acciones tomo para conseguir el mejor resultado posible.
- ❖ Además, el método Montecarlo es sencillo de hacer, y muy personalizable, se ha podido añadir ciertas modificaciones que se mencionarán en el capítulo de desarrollo y que ayudan a que el método funcione mejor.
- ❖ Es un método que nunca he utilizado, por lo que realizar el TFG con dicho método me ayudaría a comprender mejor cómo funciona el método y cuáles son sus puntos a tener en cuenta a la hora de usarlo en otros problemas.

## 4 Desarrollo

---

En este capítulo se van a mostrar el desarrollo del proyecto, es decir, todos los pasos realizados hasta llegar a los resultados obtenidos.

Empezaré contando en la sección 4.1 cómo obtenemos los datos, explicaré qué estructuras utilizamos para el manejo de estos y como hemos integrado la función de perfilado en el sistema. Luego, en la sección 4.2, hablaremos de el uso que se ha hecho de Google Cloud Platform, y, por último, en la sección 4.3, veremos cómo hemos implementado el método de Montecarlo y, más importante, daremos una visión detallada de las modificaciones que hemos realizado para optimizar el método en nuestro problema.

### ***4.1 Extracción, manejo de datos y función de perfilado***

Cualquier proyecto enfocado al análisis de datos empieza por la extracción y el manejo de estos, y no va a ser distinto en nuestro caso.

Antes de empezar a desarrollar el método de Montecarlo, se ha trabajado en un módulo de manejo de datos donde se extraerán todos los datos necesarios para realizar la simulación.

El primer paso para que funcionara el algoritmo era comprobar que se podía realizar un perfilado de los demandantes, ya que se necesitaba tener sus valores originales antes de someter a los demandantes a realizar las acciones.

Para realizar el nuevo perfilado del demandante, primero se ha tenido que extraer los datos del fichero de demandantes.

Para extraer dichos datos se ha utilizado la clase `csv.reader`, que contiene los métodos necesarios para recorrer el fichero fila a fila leyendo los datos.

Todos estos datos se han guardado en una matriz numpy preinicializada para poder utilizarlos de manera sencilla y eficaz en nuestro perfilado. La matriz numpy contendrá todos los demandantes que se pidan en las filas, y las columnas representarán las variables de los demandantes.

Se hace esto mismo con la matriz de coeficientes, para obtener una matriz numpy con los coeficientes de las variables para cada horizonte temporal (Ver referencia [20])

Uno de los problemas que hay, es que para realizar el perfilado hay que multiplicar escalarmente la matriz de demandantes por la matriz de coeficientes, por lo tanto, las matrices tienen que tener dimensiones de:

$M \times N$  (Matriz de demandantes) y  $N \times K$  (Matriz de Coeficientes)

Es decir, la primera matriz tiene tantas  $N$  columnas como  $N$  filas tiene la segunda.

Sin embargo, las dimensiones de ambas matrices no coinciden con la regla requerida, esto es porque la matriz de coeficientes solo tiene las variables más significativas. Esto significa que hay que reducir la matriz de variables del demandante, para que solo contenga aquellas variables significativas que contiene la matriz de coeficientes.

Con la matriz reducida, ya se puede multiplicar vectorialmente los vectores de demandantes y los coeficientes correspondientes al horizonte temporal que se esté hallando para obtener los valores de las probabilidades sin normalizar.

Una vez obtenidos los valores sin normalizar de cada horizonte temporal, solo queda normalizarlos para obtener las probabilidades y los segmentos correspondientes según los umbrales de los horizontes temporales.

Una vez se ha obtenido el perfilado de todos los demandantes, se escriben los resultados en un fichero para poder usarlos en el futuro sin necesidad de realizar todos los pasos anteriores.

Es importante mencionar que la obtención de la matriz de demandantes ya perfilados se puede obtener de dos maneras:

1. Leyendo el fichero desde cero, realizando el perfilado de todos los demandantes de nuevo. Esta opción es requerida en caso de querer introducir en la matriz numpy (La cual se guarda en memoria), más perfilados de demandantes.
2. Leyendo el fichero de demandantes ya perfilados con anterioridad.

Siendo la segunda opción la más óptima puesto que no tenemos que leer el fichero completo. También podemos elegir el número de demandantes a leer. Lo que nos servirá para más tarde realizar pruebas sin necesidad de leer todos los demandantes del fichero.

## **4.2 Google Cloud Platform**

Después de realizar las funciones de extracción de datos y perfilado, surgieron problemas al intentar cargar toda la matriz numpy de demandantes en memoria, ya que esta era demasiado grande.

Para solventar este problema, se ha optado por usar la plataforma de Google GCP, la cual, como ya hemos dicho, nos permite usar una máquina virtual con más memoria y CPU's.

Para poder usar los servicios que presta GCP, primero se pidió acceso para entrar en una instancia (ya creada para el proyecto) con la cuenta que tenía yo.

Después, se crearon un par de claves pública y privada para garantizar la seguridad de la instancia, de manera que solo pueda acceder el perfil confirmado, teniendo la clave privada.

Existe la posibilidad de personalizar la máquina virtual, en nuestro caso, para cargar la tabla utilizamos una máquina con sistema operativo Ubuntu de 2 CPUs y 16 GB de memoria RAM.

Ya configurada la máquina, se puede acceder desde la plataforma. Como el sistema operativo era Ubuntu, solo tuve que instalar las librerías necesarias para ejecutar el código.

Para la descarga y subida de archivos se utiliza el programa Filezilla, el cual permite conectarse a la máquina en remoto para el envío y descarga de archivos desde tu máquina a la máquina en remoto.

## **4.3 Método de Montecarlo**

Una vez se han obtenido los datos en matrices numpy fáciles de manejar, se ha procedido a desarrollar el algoritmo que previamente se ha analizado que podría ser una muy buena solución del problema.

A continuación, vamos a ver como se han plasmado los conocimientos teóricos del método de Montecarlo (Para entender los conocimientos teóricos sobre el árbol de Montecarlo es necesario leer el Anexo A) sobre el código para solventar el problema. Se pasarán por todos los aspectos clave del método, así como se mencionarán las peculiaridades y se explicará por que hemos hecho cada cosa.

El flujo de desarrollo del método comienza con la creación del árbol que manejará información como los itinerarios, la puntuación de cada nodo... después se ha codificado



los pasos del método, empezando por selección, luego expansión, simulación (para la cuál se ha diseñado la función que indicará la puntuación de cada nodo) y back-propagation. Por último, se ha determinado la manera de obtener el mejor nodo y se ha impuesto un umbral temporal para la elección del hijo que mejor se considere siguiendo unos criterios que se verán a continuación.

### 4.3.1 Árbol

Como hemos dicho, el desarrollo del método ha comenzado por la elaboración de un módulo que contendrá el árbol que manejará el método para descubrir el itinerario de acciones mejor.

Un árbol es un tipo abstracto de dato de estructura jerárquica.

Los árboles se componen por distintos nodos que pueden contener diferentes campos como valor, nombre, padre, profundidad...

La estructura de un árbol se compone de un nodo raíz que contendrá a su vez más subramas de nodos formando una estructura jerárquica como ya hemos dicho anteriormente.

A continuación, se van a mostrar los campos principales con los que cuenta el árbol y para que se utilizan; y después se mostrará cómo se ha adaptado el árbol para el algoritmo implementado en la simulación de Montecarlo.

Variable	Descripción
<b>Nombre</b>	Id del nodo. Se corresponde con la acción a realizar sobre el demandante.
<b>Puntuación</b>	Puntuación del nodo. Nos sirve para comparar la mejora de un nodo con respecto a otro.
<b>NVisitas</b>	Número de visitas realizadas al nodo. La selección y la puntuación de un nodo dependen de este campo.
<b>Hijos</b>	Hijos del nodo. Son acciones posibles que se pueden tomar después de realizar la opción del nodo.
<b>Padre</b>	Padre del nodo. Este link es necesario para poder pasar la información hacia arriba en la fase de Back-Propagation.
<b>tiempoAcc</b>	Tiempo acumulado. Esta variable lleva control del tiempo

	en meses que llevaría en la vida real realizar las acciones de la rama que recorre desde el nodo hasta la raíz.
<b>Profundidad</b>	Profundidad del nodo. Nos diferencia de un nodo u otro con el mismo nombre. También muestra la profundidad alcanzada por el árbol.
<b>tiempoMáximo</b>	Tiempo Máximo de cada nodo. El tiempo acumulado de la subrama desde el nodo a la raíz nunca va a superar el tiempo máximo.
<b>pool</b>	Opciones posibles. Opciones disponibles de la rama.

**Tabla 4-3-1: Tabla de campos del árbol. Podemos ver las variables de la clase nodo junto con una breve descripción.**

A la hora de desarrollar el método de Montecarlo, se ha pensado que sería menos costoso tratar de quitarle complejidad al módulo que contiene el método de Montecarlo. Se quería conseguir que la manera de recorrer el árbol, añadir o quitar acciones posibles y controlar la temporalidad no fuese un peso para el método, de manera que se ha decidido incluir todas las funciones encargadas de ello en la clase que contiene los nodos.

Dicho esto, a continuación, se van a mostrar lo que se ha añadido al módulo nodo para manejar las acciones del árbol:

- ❖ En primer lugar, el nodo raíz tiene que inicializar su pool de acciones posibles, de manera que, al expandir dicho nodo, tendrá todas las acciones cuyo tiempo de realización sea menor que el tiempo máximo del nodo. Además, es necesario tener en cuenta que acciones se pueden realizar para los demandantes, ya que un demandante que tiene el nivel de inglés bajo, solo podrá aumentar el nivel de inglés al nivel medio dentro de dicho pool de opciones.
- ❖ Al añadir un hijo al nodo, este hijo inicializará el padre con el nodo al que está siendo añadido, la profundidad a la del nodo padre más uno, y el tiempo acumulado al del padre más el tiempo que requiera realizar la acción que se haga en el nodo hijo.
- ❖ La característica más importante del árbol es la capacidad de manejar el pool de acciones posibles para cada nodo, esto significa que el nodo hijo heredará un pool de opciones que dependerá de las acciones realizadas anteriormente. Por

un lado, hay que tener en cuenta las acciones que se eliminan, ya que, si el padre ha realizado dicha acción, esta no debería de aparecer en el pool del hijo. Y, por otro lado, hay que añadir aquellas acciones que ya se puedan realizar como consecuencia de la acción realizada por el padre, para que se entienda mejor, pongamos como ejemplo que el padre tiene como acción aumentar el nivel de ingles a nivel medio, ahora el hijo podrá incluir la opción de aumentar nivel de inglés a nivel alto.

#### 4.3.2 Selección

Ya que se cuenta con la estructura de datos que el método va a manejar, se puede empezar a codificar el método en sí. Para ello empezaremos por su primer paso, que se trata de la selección del nodo.

En la selección hemos utilizado la siguiente heurística:

$$v_i + C \times \sqrt{\frac{\ln N}{n_i}}$$

**Figura 4-3-3-1: Heurística de la fase de selección**

Donde:

- ❖  $V_i$ : puntuación del nodo
- ❖  $C$ : Constante
- ❖  $N$ : n° visitas del padre
- ❖  $N_i$ : visitas del nodo

Esta heurística valora los nodos que tengan más puntuación, que hayan sido visitados menos veces y aquellos cuyos padres hayan sido visitados más veces. De este modo nos podemos asegurar de que el árbol se expanda descubriendo todas las posibilidades que tenemos, pero que de alguna manera premie aquellos caminos que tengan mejor puntuación.

### 4.3.3 Expansión

El siguiente paso después de la selección de un nodo es la expansión de este.

Ya se ha contado anteriormente que los árboles tienen un atributo llamado pool el cuál se trata una lista de opciones posibles de entre todas las acciones para expandir el nodo.

En la expansión se obtiene dicho pool que contiene cada una de las posibles acciones, y después se le insertan al padre todos los nodos inicializados con los nombres del pool obtenido. Aparte de esto cada vez que se crea un hijo, este contendrá una puntuación inicial, que será diferente dependiendo de la clase del padre en el momento de la creación.

Para verlo más claro, en la siguiente tabla se pueden ver los posibles casos a la hora de crear las puntuaciones de los hijos.

Clase del Padre	Puntuación inicial del hijo
D	0
C	0.25
B	0.5
A	0.75

Tabla 4-3-2: Tabla de puntuaciones iniciales de los hijos según el grupo del padre.

### 4.3.4 Puntuación

Para continuar con el paso de simulación de los nodos, primero debemos diseñar que función se utilizará para representar la puntuación de cada nodo.

La puntuación es un valor que permite determinar el desempeño que tiene un nodo con respecto a otro, a mejor puntuación, el nodo va a ser mejor y, por tanto, el demandante tendrá unas mejores condiciones de desempleo al realizar el itinerario de acciones propuesto por dicho nodo.

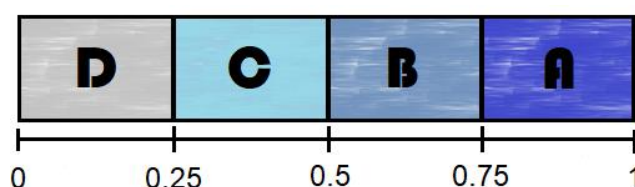
Es importante elegir bien la función que determinará el valor de la puntuación de cada nodo, pues como se ve en el Anexo A, la heurística utilizada en la selección de los nodos tiene muy en cuenta la puntuación de estos, es decir, cuanto más alta sea dicha puntuación, más probabilidad tendrán de ser elegidos.

El algoritmo que calcula la puntuación de los nodos para este proyecto tiene que tener en cuenta 2 cosas principalmente, la primera es que diferencie entre grupos, un nodo en el que las simulaciones realizadas tengan como resultado medio que el demandante ha aumentado de grupo, siempre va a ser mejor que un nodo en el que no haya aumentado.

Y, por otro lado, también tiene que valorar dentro de los nodos cuyas simulaciones tengan como resultado que el demandante acabe en el mismo grupo que los demás, aquel que tenga mayor probabilidad de salir del desempleo en el horizonte temporal que le corresponda a cada grupo (Para los grupos A y B: 3 meses, para el grupo C: 6 meses y para el grupo D: 12 meses).

Es por estas razones que se ha decidido segmentar la puntuación en 4 cuartiles diferentes, el primer cuartil le corresponderá al grupo D, el segundo cuartil al grupo C, el tercero al grupo B y el último al grupo A. De esta manera se diferencia entre los nodos de distintos grupos ya que el nodo que acabe en el grupo C es mayor siempre que un nodo en el grupo D.

Un nodo no visitado tendrá la puntuación elegida en la expansión y por tanto su grupo será el correspondiente al cuartil en el que esta se encuentre, sin embargo, a medida que este nodo es visitado, la puntuación cambiará debido a la simulación y a la propagación de la puntuación como veremos en las siguientes subsecciones.



**Figura 4-3-2:** Esta figura representa los segmentos en los que se divide la puntuación de los nodos, como vemos cada segmento comprende una longitud de 0.25 y a medida que avanza el grupo la puntuación será mayor, de esta manera podemos diferenciar los grupos en la score.

Como hemos visto en la sección 2.1.2, existen cuatro grupos diferentes según el perfil, para diferenciarlos hemos utilizado la nomenclatura de la figura, de manera que, D se corresponde con el grupo con menos probabilidad de salir del desempleo, y cuyo horizonte temporal a tener en cuenta será el de 12 meses, y a medida que aumentamos de grupo, las probabilidades de salir del desempleo serán más altas, y el horizonte temporal más bajo, hasta llegar al grupo A, cuyo horizonte temporal será de 3 meses. en

### 4.3.5 Simulación

La simulación es la parte que más cambia del método para todos los casos, pues cada problema tiene una manera de simular y obtener una puntuación diferente.

Para entender todo de una manera más clara, a continuación, se listarán los pasos a realizar en la simulación para obtener la puntuación del nodo.

- ❖ En primer lugar, es necesario iterar la simulación que realicemos con un bucle que recorra la simulación tantas veces como se le indique en el número de simulaciones.
- ❖ A continuación, se obtiene una lista de las acciones existentes en la rama que vamos a simular.
- ❖ Para la simulación se añaden más atributos al demandante de manera aleatoria hasta completar el tiempo máximo. Para esto se debe hacer una copia del demandante de manera que, al modificar dicha copia, el demandante original conserve sus valores.
- ❖ Una vez copiado el demandante, se añaden varios nodos mediante la función de añadir nodos definida en la clase. Se utiliza la función del nodo para que el pool de cada nodo hijo de la simulación cambie de acciones cada vez que se añada un nodo hijo.
- ❖ Con los atributos añadidos en la rama, se realizan todas las acciones al demandante, esto significa que se cambian los valores de las variables que dependan de las acciones realizadas.
- ❖ Hecho lo anterior, ya se obtiene el perfilado del demandante del que se obtienen las nuevas probabilidades y el nuevo grupo.
- ❖ Teniendo las probabilidades y el grupo, se puede obtener la puntuación de cada simulación de la siguiente forma:
  - Si el grupo nuevo pertenece al grupo D, la puntuación será  $0.25 * p_{12'm}$
  - Si el grupo nuevo pertenece al grupo C, la puntuación será  $0.25 + 0.25 * p_{6'm}$

- Si el grupo nuevo pertenece al grupo B, la puntuación será  $0.5 + 0.25 * p3'm$
- Si el grupo nuevo pertenece al grupo A, la puntuación será  $0.75 + 0.25 * p3'm$
- ❖ Una vez terminada la simulación, borramos la subrama creada anteriormente de manera aleatoria, sumamos la puntuación obtenida a un acumulador y volvemos a simular con otras acciones hasta acabar el bucle.
- ❖ La puntuación final de la simulación será el acumulador de puntuación dividido entre las veces simuladas.

### 4.3.6 Back-propagation

Tras la simulación del nodo, es necesario enviar la información obtenida al resto de nodos padre de la rama de la que nuestro nodo cuelga.

La fase de back-propagation se encarga de dicha tarea, para la realización esta, se ha implementado un método recursivo que actualiza la puntuación del nodo, y se llama a sí misma con el nodo padre y la puntuación previamente calculada del hijo como argumentos.

Para actualizar la puntuación del nodo, se obtiene la puntuación de la simulación (pasada por argumento) y se suma dicha cantidad a otro acumulador de puntuación del nodo. La puntuación del nodo en ese momento será la división entre este acumulador y el número de visitas del nodo.

### 4.3.7 Obtener el mejor hijo

Ya se han iterado todas las simulaciones de Montecarlo hasta llegar al número de pasos indicado.

Ahora se tiene un árbol con muchos nodos expandidos y cada nodo tendrá una puntuación propia.

Solo queda elegir cual de todos los nodos es el mejor, y el itinerario de acciones compuesto por las acciones de cada nodo de la rama será el mejor itinerario a tomar por el demandante para que mejore su situación de empleabilidad.

#### 4.3.8 Umbral temporal

En el sistema de simulaciones se ha intentado ser muy minucioso a la hora de escoger el mejor hijo, pues esta decisión es muy importante a la hora de recomendar un itinerario u otro.

Es por ello por lo que se ha pensado en una manera de valorar positivamente el hecho de que un itinerario sea temporalmente más corto que otro. Esto se ha conseguido mediante la utilización de un umbral temporal el cual sumará un peso relativo a la diferencia temporal entre dos itinerarios para darle más prioridad a aquellos itinerarios más cortos.

Para explicarlo mejor, a continuación, se van a listar los pasos del algoritmo realizado:

- ❖ En primer lugar, se obtienen todos los nodos terminales del árbol, estos nodos son aquellos que no tienen ningún hijo.
- ❖ Después de obtener la lista de nodos terminales, se ordenan y agrupan dichos nodos en grupos según su tiempo acumulado.
- ❖ Una vez agrupados y ordenados los nodos, se escoge el mejor de ellos para cada horizonte temporal.
- ❖ Se obtienen los nuevos grupos de cada nodo en cada horizonte temporal.
- ❖ Ahora, hay que comparar de menor a mayor horizonte temporal, cada nodo mejor, de manera que si el nodo a verificar cumple las condiciones que hay que cumplir con los nodos siguientes, elegiremos dicho nodo como mejor nodo y lo devolveremos, y en caso de que lleguemos al último nodo, este será devuelto.

Dichas condiciones son:

- Que el nodo con menor temporalidad tenga un grupo igual o mejor que los nodos con mayor temporalidad
- Que la puntuación del nodo a evaluar sumado a un factor que tiene en cuenta la diferencia temporal y el umbral temporal (Elegido previamente) sea mayor que la puntuación de los nodos con mayor temporalidad.

Hay que mencionar, que cuanto mayor es dicho umbral de temporalidad, más se tiene en cuenta el factor temporal que se suma en las comprobaciones, por lo cual, es más probable que se elija un nodo con menor temporalidad.



#### **4.3.9 Últimos pasos**

Ya, una vez obtenido el mejor hijo de todos, el itinerario obtenido que maximiza los resultados en el menor tiempo posible será aquel formado por las acciones de la rama del nodo elegido.

Ya escogido el itinerario, se realiza el perfilado para obtener las probabilidades y el grupo nuevos y escribimos los resultados en un fichero csv para obtener los resultados.

## 5 Integración, pruebas y resultados

---

En este capítulo vamos a mostrar el fruto de las simulaciones de Montecarlo.

En primer lugar, vamos a explicar qué población hemos utilizado en las pruebas de validación en la sección 5.1. Después, en la sección 5.2, contaremos como hemos elegido los hiper parámetros para la ejecución del algoritmo, y finalmente, en la sección 5.3 se mostrarán las pruebas que se han realizado junto con los resultados obtenidos y el significado de dichos resultados.

### ***5.1 Elección de demandantes para la validación***

Este es un paso importante para la obtención de los resultados.

Se ha tenido que reducir la población a una muestra significativa de demandantes debido a que, el método de Montecarlo al aumentar el número de pasos y de simulaciones, escala linealmente tanto en tiempo (llevando más tiempo realizar la simulación de un demandante), como en calidad de los resultados (a más iteraciones, mejores son los resultados obtenidos).

Para reducir la población se ha tenido en cuenta el tiempo de proceso del algoritmo, siempre que la calidad de los resultados muestre la potencia que tiene el método a la hora de escoger itinerarios óptimos.

Para escoger la población de prueba se ha realizado un método que elige una población aleatoria de 50.000 demandantes de entre el total de demandantes del fichero, en el que cada demandante es único. Para ello se han obtenido índices aleatorios utilizando la librería “random.py” [8] que serán las filas leídas en el fichero de demandantes total.

### ***5.2 Elección de hiper parámetros***

La elección de hiper parámetros se ha realizado mediante un proceso deductivo en el que se comparan diferentes combinaciones posibles de hiper parámetros.

En primer lugar, se ha pensado como afecta cada hiper parámetro en los resultados de la simulación.

- ❖ **C:** es la constante de la heurística de elección de nodos. Como se puede ver en la **Figura 4-3-X**, C multiplica al factor que relaciona al número de visitas del padre, con el número de visitas del nodo hijo, esto significa que, a mayor C, mayor importancia le damos a dicho factor, tendiendo a expandir más el árbol sin tener demasiado en cuenta la puntuación de cada nodo.
- ❖ **Número de pasos:** Es el número de simulaciones de Montecarlo, es decir, el número de veces que se va a elegir, expandir, simular y propagar la información de un nodo. El número de pasos determinará el número de nodos descubiertos, esto significa que dicho número dependerá directamente del factor de ramificación del árbol.
- ❖ **Número de simulaciones:** Es el número de veces que un nodo se va a simular en la etapa de simulación. Cuantas más simulaciones haya por nodo, más precisa va a ser la puntuación representativa de cada nodo, y, por tanto, más precisa será nuestra elección de los nodos.

Para C se ha elegido el valor 1.4 ya que es un factor que suele ser utilizado con el método de Montecarlo que funciona bien en el nuevo sistema.

Al umbral de temporalidad se le dará un valor de 0.001. Es un valor bajo para no afectar a la eficacia de los itinerarios, aunque es un valor justo para dar preferencia a los itinerarios más cortos en caso de valores extremadamente parecidos.

En un futuro convendría cambiar dicho valor en función a la valoración que aporte la Comunidad de Madrid.

Cabe destacar que las pruebas realizadas se harán para un horizonte temporal de 18 meses de manera que se puedan observar diferentes itinerarios de diferente longitud temporal.

El número de pasos tiene que ser grande, ya que, el factor de ramificación del árbol es muy grande, en concreto tendrá un factor de ramificación del orden de 40, esto significa que, si el número de pasos es pequeño, no se lograrán descubrir muchas posibilidades del árbol.

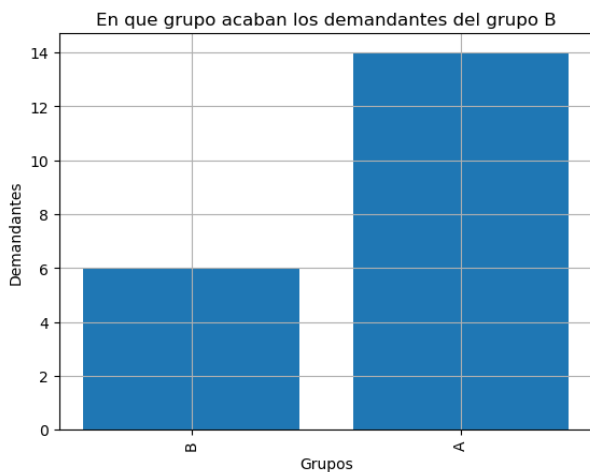
El número de simulaciones por tanto tiene que ser alto también, puesto que la selección de los nodos tiene que ser lo más precisa posible para que se puedan descubrir los mejores caminos para nuestro demandante.

Por otro lado, el tiempo de ejecución de cada simulación va a aumentar conforme al número de pasos y de simulaciones que hagamos, por lo que hay que elegir un número de pasos y de simulaciones lo suficientemente grande para que sea potente, pero lo justo para poder simular una muestra representativa de los demandantes, que en este caso será de 50.000 demandantes.

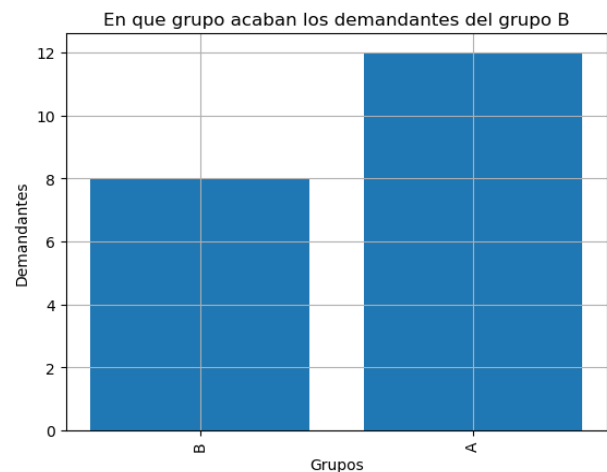
Es por ello por lo que hemos realizado diferentes pruebas con combinaciones de distintos hiper parámetros.

De estas pruebas se muestran a continuación las gráficas más significativas a la hora de elegir los hiper parámetros. Compararemos los resultados con 100 pasos, 100 simulaciones y 18 meses frente a los resultados obtenidos con 200 pasos, 200 simulaciones y 18 meses para una muestra de 1200 demandantes.

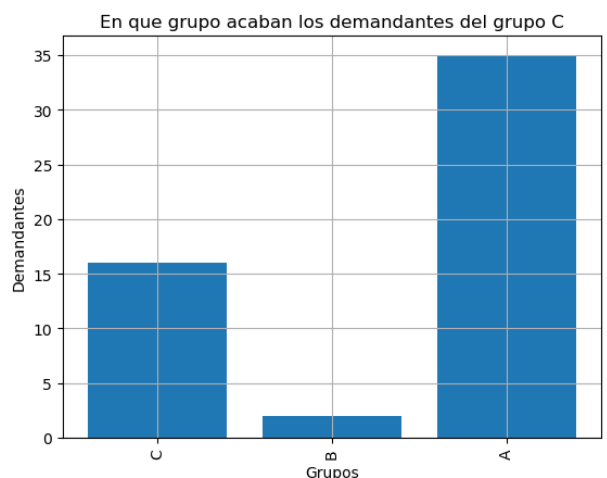
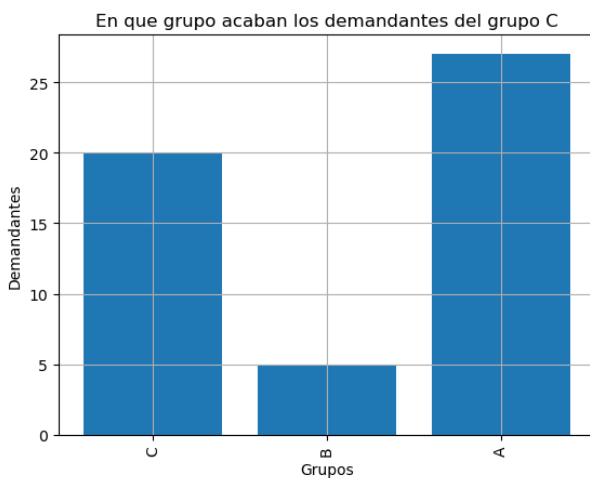
#### 100 pasos, 100 simulaciones y 18 meses



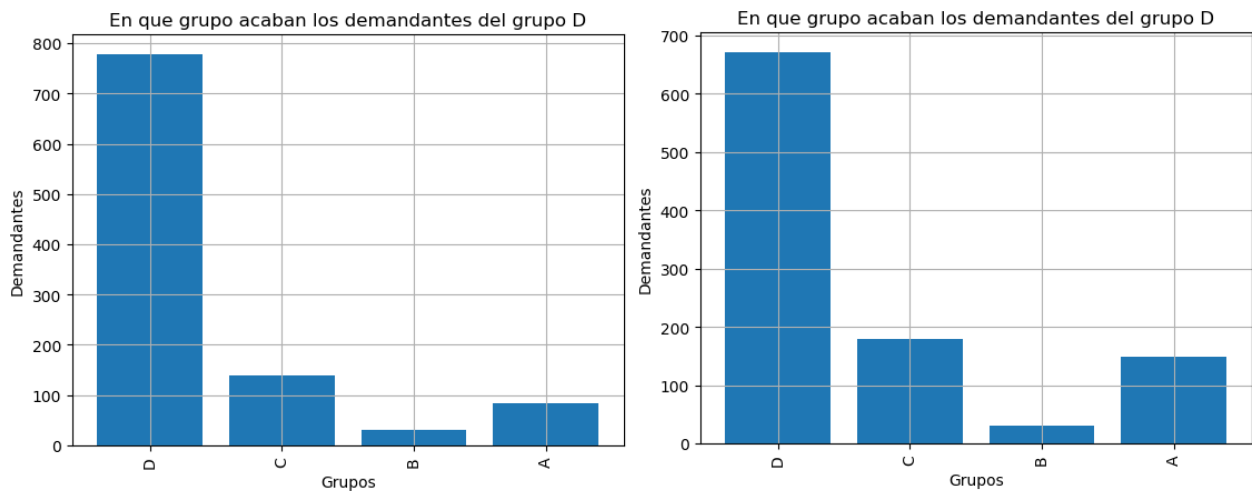
#### 200 pasos, 200 simulaciones y 18 meses



**Figura 5-2-1: Histograma del número de demandantes que pasan del grupo B a grupos superiores. Uno para cada prueba con distintos parámetros.**

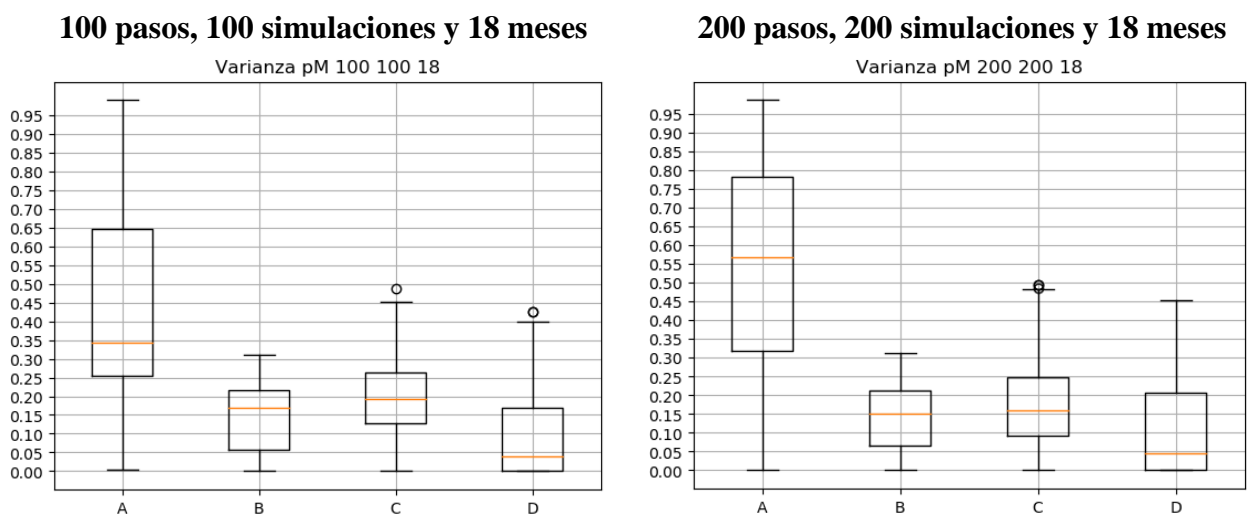


**Figura 5-2-2: Histograma del número de demandantes que pasan del grupo C a grupos superiores. Uno para cada prueba con distintos parámetros.**



**Figura 5-2-3: Histograma del número de demandantes que pasan del grupo C a grupos superiores. Uno para cada prueba con distintos parámetros.**

En las gráficas representadas en la Figura 5-2-1, vemos que, para 100 pasos y 100 simulaciones, hay dos demandantes más que aumentan de grupo, sin embargo, en la figura 5-2-2 y 5-2-3 se puede observar que hay muchos más demandantes que han cambiado de grupo y en concreto muchos demandantes que acaban en el mejor grupo (grupo A). Esto se puede apreciar muy bien en la figura 5-2-3, ya que hay una diferencia de más de 100 personas en el grupo D en la gráfica de la izquierda, y se puede observar que muchas de esas personas acaban en el grupo C y mayoritariamente en el A.



**Figura 5-2-4: Diagrama de cajas y bigotes que refleja la varianza de la probabilidad de salir del desempleo para cada grupo. Uno para cada prueba con distintos parámetros.**

Ahora viendo la figura 5-2-4, se puede percibir que hay una gran diferencia en la varianza a favor de la gráfica de la derecha para aquellas personas que acaban en el grupo

A. Sin embargo, no podemos observar un cambio apreciable en los demás grupos. Esto se puede deber a que existe mucho más movimiento de grupos bajos al grupo A, como hemos podido ver anteriormente.

Es por estas razones por lo que se va a considerar utilizar finalmente 200 pasos y 200 simulaciones para la obtención de resultados final.

### **5.3 Resultados**

En este apartado se va a comparar el sistema de simulaciones mediante el método de Montecarlo frente al antiguo método de NBA mediante diferentes gráficas que aportarán información relevante a la hora de comprobar la eficacia de ambos métodos.

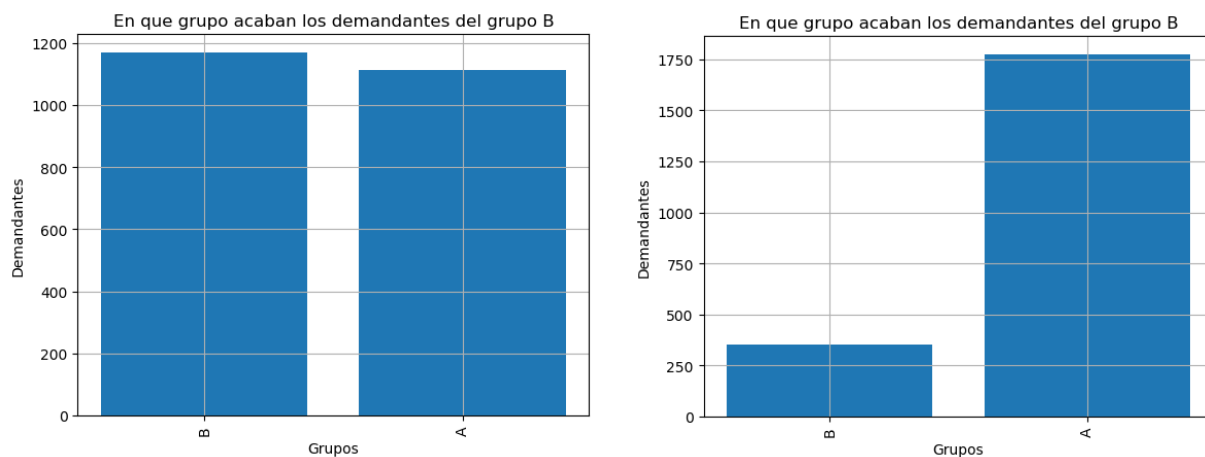
En primer lugar se va a ver el flujo de demandantes que terminan en grupos superiores al suyo, luego se comprobará la varianza de la probabilidad nueva con respecto de la antigua para cada grupo, en tercer lugar se va a comprobar que el número de acciones de los itinerarios en el método de Montecarlo varía, y a continuación veremos los meses de duración que tardaría el demandante en realizar dichos itinerarios de acciones, por último, se analizarán los itinerarios de acciones más frecuentes para cada método.

Después de analizar cada apartado de gráficas daremos una conclusión sobre los resultados.

#### **5.3.1 Flujo de demandantes**

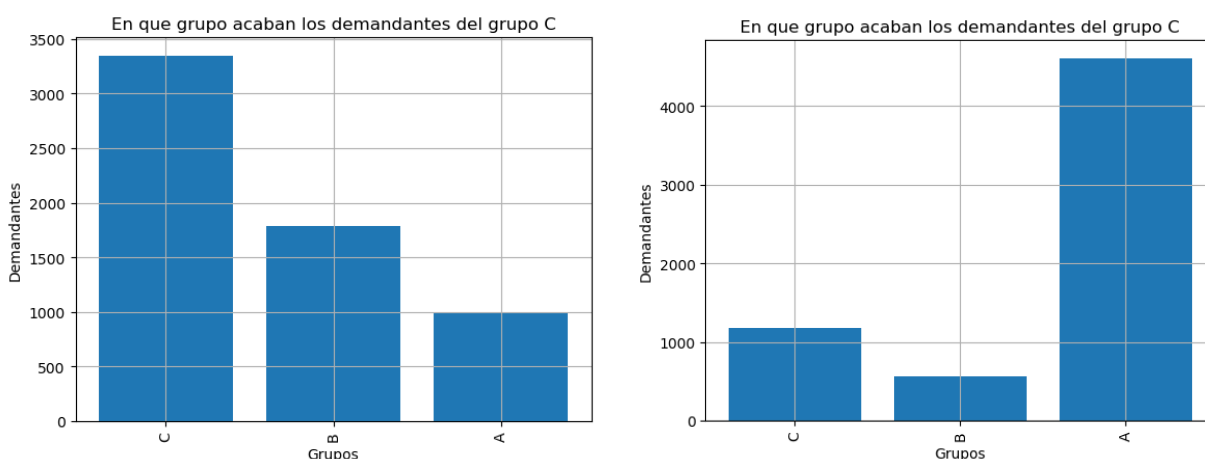
En este apartado se puede observar el flujo de demandantes que se mueven de un grupo a uno superior.

El mejor caso que se puede dar en un demandante es que aumente de grupo a uno superior, por lo que estas gráficas van a significar mucho a la hora de evaluar y comparar los resultados. A continuación, vamos a ver las gráficas comparativas del flujo de demandantes para el método de NBA (a la izquierda) y el algoritmo de Montecarlo (a la derecha)



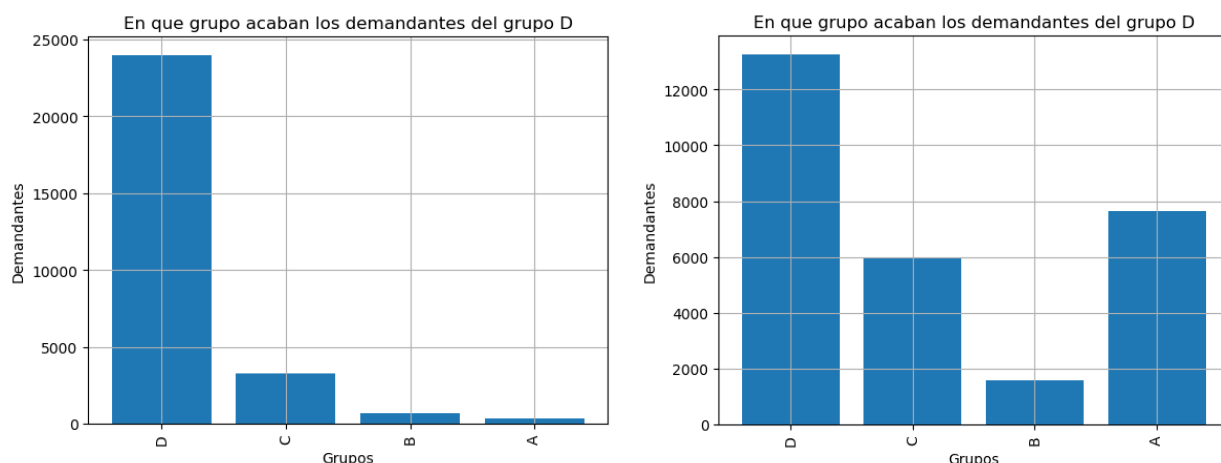
**Figura 5-3-1: Flujo de demandantes del grupo B a grupos superiores**

Se puede observar una clara diferencia en las gráficas, pues en la gráfica que representa el método NBA, hay muchos más grupos que no fluctúan hacia el grupo A. Sin embargo, la gráfica que representa el método de Montecarlo es mucho más abrupta, hay muchos más cambios de grupo lo cual es muy bueno.



**Figura 5-3-2: Flujo de demandantes del grupo C a grupos superiores**

En el caso del grupo B se puede observar también que hay muchos más movimientos a grupos superiores en las gráficas del método de Montecarlo. Además, hay que puntualizar que una enorme mayoría de demandantes vuelven a acabar en el grupo A, no como en la gráfica del método NBA, en la que la mayoría no se mueve de grupo, y el resto suele acabar en el grupo B.



**Figura 5-3-3: Flujo de demandantes del grupo D a grupos superiores**

El grupo D es el grupo más complicado de cambiar, de hecho representa una mayoría de los desempleados.

En esta gráfica se puede ver que ninguno de los dos métodos consigue buenos resultados a la hora de cambiar de grupo a los demandantes del grupo D, esto es porque hay muchos de ellos que no se contemplan, y otros que simplemente tienen una probabilidad muy baja la cual es difícil de aumentar.

Si miramos la gráfica de Montecarlo, podemos ver aun así que de casi 25000 demandantes que acaban en el grupo D con NBA, solo 13000 continúan en dicho grupo, el resto ha fluctuado a grupos superiores dando lugar a que más de la mitad de los demandantes que estaban en el grupo D hayan fluctuado a grupos superiores.

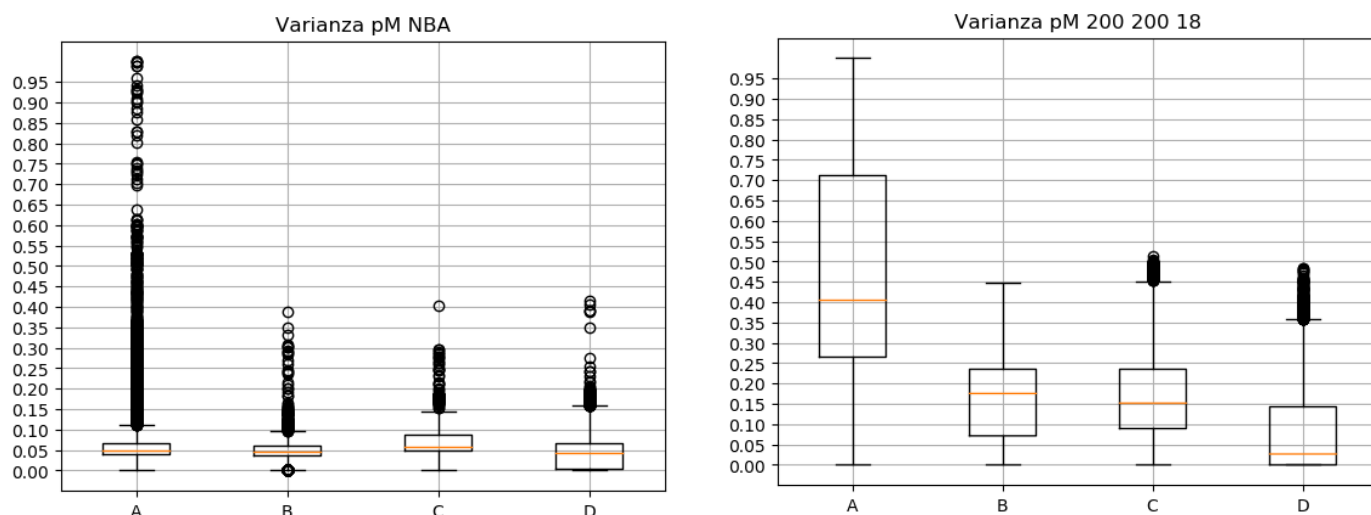
De nuevo, es muy bueno saber que de los demandantes que han subido de grupo, casi la mitad de ellos han acabado en el grupo A, lo cual garantiza una muy buena actuación del método de Montecarlo con respecto al método NBA.

Se puede concluir en que el método de Montecarlo ha ayudado a la mayoría de los demandantes a subir de grupo mientras que en NBA la mayoría de ellos se han quedado en sus respectivos grupos, esto genera una gran ventaja en el método nuevo con respecto al NBA, ya que como hemos dicho antes, la fluctuación de grupos inferiores a grupos superiores es muy importante.



### 5.3.2 Varianza de probabilidades

En esta sección se va a ver la comparativa entre las varianzas de los grupos de ambos métodos.



**Figura 5-3-4: Diagramas de cajas y bigotes que representan la varianza de la probabilidad en el horizonte temporal respectivo de cada grupo.**

En la gráfica de la izquierda, se puede observar que la media de varianza entre probabilidad simulada y original de cada grupo ronda el valor de 0,05. Además, los límites superiores e inferiores de las cajas son bastante próximos a la media. Hay que decir también que existen muchos valores atípicos en el grupo A con resultados muy positivos en la varianza, llegando a ser casi de valor 1 en unos pocos casos.

Hablando en términos más claros, según el análisis de la gráfica de la izquierda, podemos concluir que la media de demandantes de cada grupo no varía en exceso la probabilidad que tendrían al realizar las acciones indicadas, dentro del horizonte temporal indicado. Sin embargo, hay casos en los que la probabilidad si varía, pero estos casos son tan pocos que se consideran atípicos, estos valores están representados en la gráfica con forma de circunferencias

Sin embargo, en la gráfica de la varianza para el método de Montecarlo, los valores medios son mucho mayores exceptuando en el grupo D y sobre todo en el grupo A. Las cajas son mucho más amplias y los bigotes también.

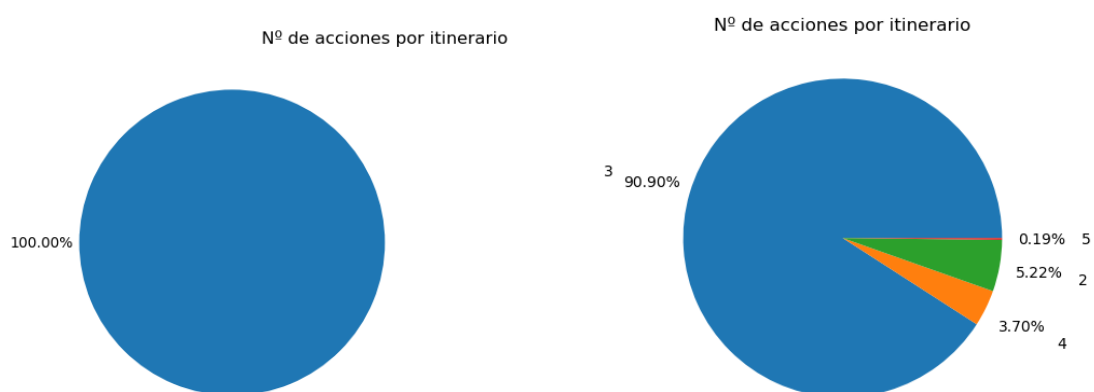
Esto anterior está indicando que las varianzas de probabilidades son mucho mayores de media para todos los grupos exceptuando el D, además podemos decir que los

valores atípicos del método NBA, son mucho más frecuentes. Podemos decir que la distribución de valores abarca un espacio más amplio en general y los valores en promedio son mucho más altos.

Para concluir, no podemos dejar escapar unos pequeños detalles. En primer lugar, se observa que la varianza máxima en los grupos B, C Y D no supera de cierto valor en ambas gráficas. Esto se debe a que cuando un demandante de cualquiera de esos grupos supera cierto umbral en su horizonte temporal, este cambia de grupo, y en dicho caso solo tendremos en cuenta la varianza de la probabilidad correspondiente al horizonte temporal del nuevo grupo. Es por esta misma razón que en ambas partes, en el grupo A se pueden observar muchos valores con varianzas más amplias, puesto que no tienen un umbral que superar. Y en segundo lugar, ya se ha podido observar que ay muchos más demandantes que suben al grupo A con el método de Montecarlo, esto hace que en dicho grupo, el método de Montecarlo, tenga mejores resultados que el de NBA.

### 5.3.3 Número de acciones por itinerario

En esta sección mostraremos el porcentaje de número de acciones por cada itinerario para mostrar que el nuevo método es capaz de recomendar itinerarios de dimensiones diferentes.



**Figura 5-3-5: Diagrama circular que representa los porcentajes de acciones por itinerario recomendado en cada caso.**

En las gráficas mostradas se puede observar que en el método NBA solo puede generar itinerarios de longitud 1, puesto a que solo recomiendan una acción, mientras que el nuevo sistema es capaz de recomendarte itinerarios más completos con más acciones a realizar por cada demandante.

Con estos datos, se quiere reflejar que en ciertas ocasiones tiene más sentido ofertar un rango de acciones más amplio en función a tener mejores resultados.

Se puede ver también que el número de acciones a tomar más probable es 3, con una mayoría de casi el 91%. Esto se puede deber a que las ramas del árbol usualmente se quedan en profundidad 3, o que existen combinaciones de 3 acciones que son mucho más eficaces que el resto de itinerarios con más acciones.

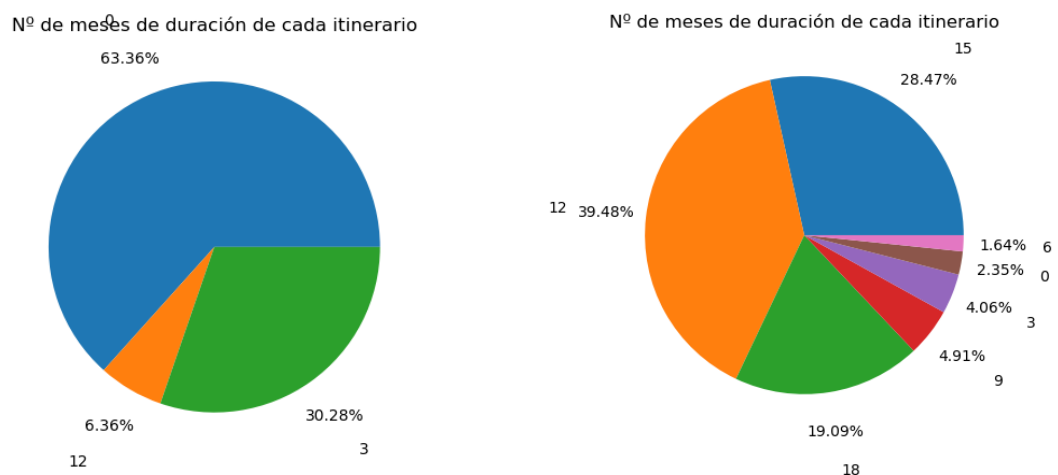
### 5.3.4 Duración de itinerarios

Ya mostrados el número de acciones a tomar por cada itinerario, es interesante mostrar también el rango de duraciones de cada itinerario ofertado y la probabilidad de estos.

La duración de los itinerarios es la suma de la duración de cada acción incluida en cada itinerario.

Los valores asignados para cada acción son:

- ❖ Solicitar una ocupación: 0 meses
- ❖ Aumentar el nivel de inglés o de informática: 3 meses
- ❖ Aumentar el nivel de formación: 6 meses
- ❖ Aumentar la experiencia en una ocupación: 12 meses



**Figura 5-3-6: Diagramas circulares con los porcentajes de la duración de los itinerarios recomendados.**

En la figura de la izquierda podemos ver que la duración que predomina es de 0 meses con un 63%, esto significa que lo más recomendado será solicitar una ocupación.

Por lo general, la duración de los itinerarios es de 3 meses, y solo en un 6% de los casos se recomendará aumentar la experiencia en una ocupación.

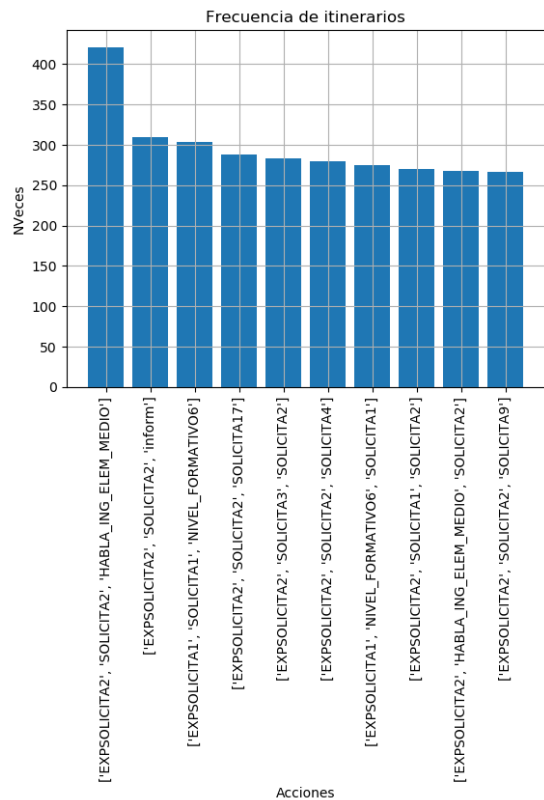
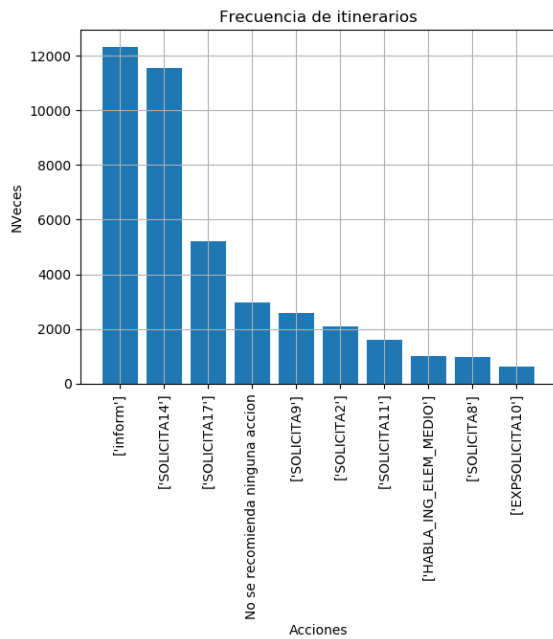
En la gráfica de la derecha no se puede saber de forma clara que se está recomendando en cada periodo, pues puede haber diversas combinaciones, sin embargo, se puede observar que los itinerarios más frecuentes tienen una duración mayor a 12 meses.

De hecho, el valor más alto de duración serían 12 meses con casi un 40%.

Estos valores se pueden ajustar si aumentamos el umbral de temporalidad, pues se escogerían más itinerarios más cortos.

### **5.3.5 Frecuencia de itinerarios**

Puede ser interesante también por último comprobar cual es el top 10 de itinerarios más frecuentes de cada método, a la izquierda se pueden ver las acciones más recomendadas por el NBA, mientras que a la derecha se ven los itinerarios más frecuentes recomendados por el método de Montecarlo.



**Figura 5-3-7: Itinerarios más frecuentes recomendados por cada método.**

De estas gráficas se puede sacar que en el método de NBA, existen dos acciones que son mucho más frecuentes que el resto, estas son informática y solicitar la ocupación 14.

Por otro lado, parece que con el sistema de Montecarlo existe un itinerario que es algo más recomendado que lo general, sin embargo, las recomendaciones de itinerarios están más repartidas, por lo que habrá más variedad de recomendaciones y permite que no se colapse tanto un servicio en concreto.

Es curioso ver también, que para el método de Montecarlo, se repiten en todos los itinerarios la acción de aumentar la experiencia en la ocupación 1 y 2, junto con la solicitud de dicha ocupación. Esto quiere decir que dicha combinación de acciones es potencialmente muy buena para los demandantes, ya que se recomienda mucho, sin embargo, con el antiguo sistema NBA, no podríamos haber llegado a esa conclusión.

## 6 Conclusiones y trabajo futuro

---

### 6.1 Conclusiones

En primer lugar, se va a hacer una conclusión sobre los resultados obtenidos en el capítulo 5.

Como se puede ver en las gráficas, los resultados del método de Montecarlo son satisfactorios.

Por una parte, se ha desarrollado un método robusto que funciona correctamente para todos los grupos y situaciones. Se ha conseguido además que el cambio de grupo en el demandante sea la acción más frecuente, lo cual significa que la mayoría de los demandantes pueden conseguir mejorar su situación de empleo en la mejor manera posible.

Además, como refleja la gráfica de la varianza, la probabilidad de cada horizonte temporal aumenta más que en el sistema anterior, de manera que, aunque el demandante no aumente de grupo, este mejore su probabilidad de encontrar trabajo en el mismo horizonte temporal.

Además, se puede observar que se han conseguido los objetivos que se habían propuesto desde un primer momento en la sección de objetivos.

1. El sistema recomienda más de una acción por itinerario como se ha visto en las gráficas. Por otro lado, este sistema incluye temporalidad de servicios, haciendo posible los itinerarios flexibles a una temporalidad determinada de X meses.
2. Se ha conseguido explorar una cantidad de opciones mayor a la que se alcanzaba con el sistema anterior, esto significa que hay una mayor variedad de itinerarios, que permitirá una mayor flexibilidad al orientador de cara a recomendar a los demandantes que salidas tiene.

A parte de todo lo dicho, se han implementado nuevas características como el umbral de temporalidad en los itinerarios que permitirá ajustar los itinerarios a una temporalidad más reducida.

Finalizando, el nuevo sistema de empleabilidad ha conseguido lograr los objetivos propuestos y se ha podido aprender sobre nuevas tecnologías y métodos de análisis nuevos en el proceso de desarrollo de este.

## **6.2 Trabajo futuro**

Hay muchas posibles mejoras para el futuro.

En primer lugar, se podría adaptar el número de iteraciones y simulaciones en función de cada demandante, de manera que, si un demandante no encuentra un itinerario que mejore su situación en un tanto por ciento, este siga buscando una solución, o en caso contrario, si un demandante encuentra una solución lo suficientemente satisfactoria, este deje de iterar para continuar computando otros demandantes.

Otra idea podría ser el desarrollo de un sistema que permita buscar servicios vigentes que permitan a los demandantes realizar las acciones que han sido recomendadas.

También se podría recomendar más de un itinerario de manera que al orientar el desempleado este tenga diferentes opciones a elegir, para encontrar una solución que se adapte a él.

Por último, sería buena idea desarrollar algún método predictivo que sea capaz de obtener el valor del umbral de temporalidad para optimizar que el demandante encuentre trabajo lo antes posible sin perder la eficacia del itinerario.

# Referencias

---

- [1] Introducción a los métodos de Monte-Carlo con Python, autor: Raúl E. López Briega, URL: <https://relopezbriega.github.io/blog/2017/01/10/introduccion-a-los-metodos-de-monte-carlo-con-python/>, 28/04/2019.
- [2] Monte Carlo Tree Search – beginners guide, autor: anónimo, URL: [https://int8.io/monte-carlo-tree-search-beginners-guide/#Monte\\_Carlo\\_Tree\\_Search\\_8211\\_basic\\_concepts](https://int8.io/monte-carlo-tree-search-beginners-guide/#Monte_Carlo_Tree_Search_8211_basic_concepts), 14/04/2019.
- [3] ¿QUÉ SON LOS MÉTODOS DE MONTECARLO?, autor: Universidad de Alcalá, URL: <https://www.master-finanzas-cuantitativas.com/metodos-de-montecarlo/>, 14/04/2019
- [4] NP – Completos Algoritmos Heurísticos, autores: David Redondo y José del Castillo Izquierdo, Fecha de edición: 2 de abril.
- [5] How to return dictionary keys as a list in Python, autor: Jim Fasarakis Hilliard, URL: <https://stackoverflow.com/questions/16819222/how-to-return-dictionary-keys-as-a-list-in-python>, 06/05/2019
- [6] Python | Sort Python Dictionaries by Key or Value, autor: Tanmay\_Jain, URL: <https://www.geeksforgeeks.org/python-sort-python-dictionaries-by-key-or-value/>, 03/05/2019
- [7] How do I sort a dictionary by value?, autor: Devin Jeanpierre, URL: <https://stackoverflow.com/questions/613183/how-do-i-sort-a-dictionary-by-value>, 03/05/2019
- [8] Itertools – Functions creating iterators for efficient looping, autor: anónimo, URL: <https://docs.python.org/3/library/itertools.html#itertools.groupby>, 15/04/2019
- [9] Csv – CSV File Reading and Writing, autor: anónimo, URL: <https://docs.python.org/2/library/csv.html>, 12/04/2019



- [10] Pandas, autor: pydata, URL: <https://pandas.pydata.org/>, 12/04/2019
- [11] When processing CSV data, how do I ignore the first line of data, autor: Martineau, URL: <https://stackoverflow.com/questions/11349333/when-processing-csv-data-how-do-i-ignore-the-first-line-of-data>, 14/04/2019
- [12] Diagrama Cajas y Bigotes, autor: anónimo, URL: [https://datavizcatalogue.com/ES/metodos/diagrama\\_cajas\\_y\\_bigotes.html](https://datavizcatalogue.com/ES/metodos/diagrama_cajas_y_bigotes.html), 04/06/2019
- [13] Python rango for float numbers with Examples, Python, autor: Vishal, URL: <https://pynative.com/python-range-for-float-numbers/>, 15/05/2019
- [14] Los Tipos Abstractos de Datos, autor: anónimo, URL: <http://webdiis.unizar.es/~elvira/eda/material0304/TADespec/TAD.pdf>, 04/05/2019
- [15] Scipy.stats.norm, autor: anónimo, URL: <https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.stats.norm.html>, 20/05/2019
- [16] Random – Generate pseudo-random numbers, autor: anónimo, URL: <https://docs.python.org/3/library/random.html>, 12/05/2019
- [17] Copy - Shallow and deep copy operations, autor: anónimo, URL: <https://docs.python.org/2/library/copy.html>, 17/05/2019
- [18] Codecs – Codec registry and base classes, autor: anónimo, URL: <https://docs.python.org/2/library/codecs.html>, 29/05/2019
- [19] random.py, URL: <https://docs.python.org/3/library/random.html>, 01/06/2019
- [20] NumPy, autor: Scipy.org, URL: <https://www.numpy.org/>, 16/04/2019
- [21] Python Statistics Tutorial | Mean Function Example, autor: Krunal, URL: <https://appdividend.com/2019/01/28/python-statistics-tutorial-mean-function-example/>, 27/05/2019
- [22] How to name the ticks in a python matplotlib boxplot, autor: anónimo, URL: <https://stats.stackexchange.com/questions/3476/how-to-name-the-ticks-in-a-python-matplotlib-boxplot>, 28/05/2019

- [23] Pylab\_examples example code: boxplot\_demo.py, autores: John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib development team; 2012 – 2016, URL: [https://matplotlib.org/examples/pylab\\_examples/boxplot\\_demo.html](https://matplotlib.org/examples/pylab_examples/boxplot_demo.html) , 28/05/2019
- [24] Time – Time access and conversions, autor: anónimo, URL: <https://docs.python.org/3/library/time.html>, 30/04/2019
- [25] Operator – Standard operators as functions, autor: anónimo, URL: <https://docs.python.org/3/library/operator.html>, 30/04/2019
- [26] Statistics – Mathematical statistics functions, autor: anónimo, URL: <https://docs.python.org/3/library/statistics.html>, 02/05/2018
- [27] Itertools – Functions creating iterators for efficient looping, autor: anónimo, URL: <https://docs.python.org/3/library/itertools.html#itertools.islice>, 12/05/2018
- [28] Gráficos Boxplot con Python, autor: Italo Farfán Vera, URL: <https://italoffvv.wordpress.com/2013/12/19/graficos-boxplot-con-python/>, 28/05/2019
- [29] How do I draw a grid onto a plot in Python?, autor: Andrey Sobolev, URL: <https://stackoverflow.com/questions/8209568/how-do-i-draw-a-grid-onto-a-plot-in-python>, 28/05/2019
- [30] Matplotlib Bar Chart, autor: anónimo, URL: <https://pythonspot.com/matplotlib-bar-chart/>, 28/05/2019
- [31] Grafico torta con matplotlib + Python, autor: anónimo, URL: <https://codigosimportantes.blogspot.com/2012/10/grafico-torta-con-matplotlib-python.html>, 02/06/2019
- [32] Matplotlib percent label position in pie chart, autor: anónimo, URL: <https://stackoverflow.com/questions/21572870/matplotlib-percent-label-position-in-pie-chart>, 29/05/2019
- [33] How to convert list to string, autor: Senthil Kumaran, URL: <https://stackoverflow.com/questions/5618878/how-to-convert-list-to-string>, 06/05/2019

- [34] Statistical functions (scipy.stats), autor: SciPy.org, URL:  
<https://docs.scipy.org/doc/scipy/reference/stats.html>, 02/05/2019
- [35] Math – Mathematical functions, autor: anónimo, URL:  
<https://docs.python.org/3/library/math.html>, 17/05/2019

## Glosario

---

NBA	Next Best Action (Mejor Acción Posible)
TAD	Tipo Abstracto de Dato
Open Source	Licencia abierta
Input	Datos de entrada
GCP	Google Cloud Platform

## **Anexos**

---

### ***A Método de Montecarlo***

Primero se va a poner un poco de contexto sobre qué es el método de Montecarlo y luego se explicará el funcionamiento de este.

En primer lugar, el método de Montecarlo es un método estadístico que se basa en la probabilidad para obtener aproximaciones de problemas difíciles de computar.

El nombre de método de Montecarlo se acuñó en referencia al Casino de Montecarlo (Mónaco), ya que en esta ciudad los juegos de azar tienen mucha importancia.

El método de Montecarlo consiste en la repetición de cuatro simples pasos:

#### **Selección**

El primer paso de una simulación de Montecarlo es la selección.

Este paso consiste en recorrer el árbol de búsqueda para seleccionar un nodo de todos aquellos nodos descubiertos.

La selección del nodo es dependiente de una heurística que debe asegurar que el árbol expanda sus ramas de manera equitativa, pero dando cierta prioridad a los nodos con mejor puntuación.

Por último, hay que mencionar que, si un nodo es seleccionado, este deberá de aumentar el número de visitas de este, esto ocurrirá también al seleccionar un nodo después de haberlo expandido en la siguiente fase.

#### **Expansión**

Después de haber seleccionado un nodo, se pasa a expandirlo.

La expansión consiste en introducir los hijos de un nodo en el conjunto de nodos visitables.

Los nodos hijo del nodo elegido tendrán una puntuación a elegir por el diseñador del algoritmo, veremos en detalle cómo se inician dichos nodos en el apartado 3.3. Una vez expandido el nodo, se selecciona uno de ellos al azar para pasar a la siguiente fase.

## **Simulación**

Ya elegido el nodo hay que realizar la simulación de este.

La simulación no es más que el cálculo de la puntuación del nodo. Para esto iteramos  $n$  veces el siguiente algoritmo:

Primero se elige una subrama del nodo de manera aleatoria.

Después se calcula la puntuación de la rama y se suma a un acumulador del nodo que está siendo simulando.

Una vez iterado el algoritmo anterior, se realiza la media de dichas simulaciones para obtener la puntuación de la simulación.

La puntuación del nodo será la media entre todas las puntuaciones de las simulaciones realizadas dividido entre el número de visitas del nodo simulado.

## **Back-propagation**

Una vez realizada la simulación del nodo ya solo queda el último paso, hay que hacer que los padres del nodo se enteren.

Para ello, simplemente hay que propagar la información de los hijos hacia los padres, cuando el padre recibe la puntuación de su hijo, este tiene que actualizar su puntuación haciendo la media entre la puntuación acumulada y el número de visitas del nodo padre



